NAME: _____

SIGNATURE: _____

STUDENT #: _____

SECTION: _____

UNIVERSITY OF SASKATCHEWAN
Department of Computational Science

CMPT 111.3
FINAL EXAMINATION
December 12, 1994

Total Marks: 100   Time: 3 Hours

OPEN BOOK

This exam consists of short answer questions and programming problems. Read each question carefully and write your answer legibly **on this question sheet**. You may use the backs of pages or answer books for rough work but all final answers must be on the question sheet. The marks for each question are indicated in the margin. Allocate your time accordingly. Ensure that your **name** AND **student number** are clearly written on the examination paper and that your name is on every page. When you have finished the exam, hand in the examination paper and ALL answer books to your instructor.

Marks

15   1.   Crossword Puzzle
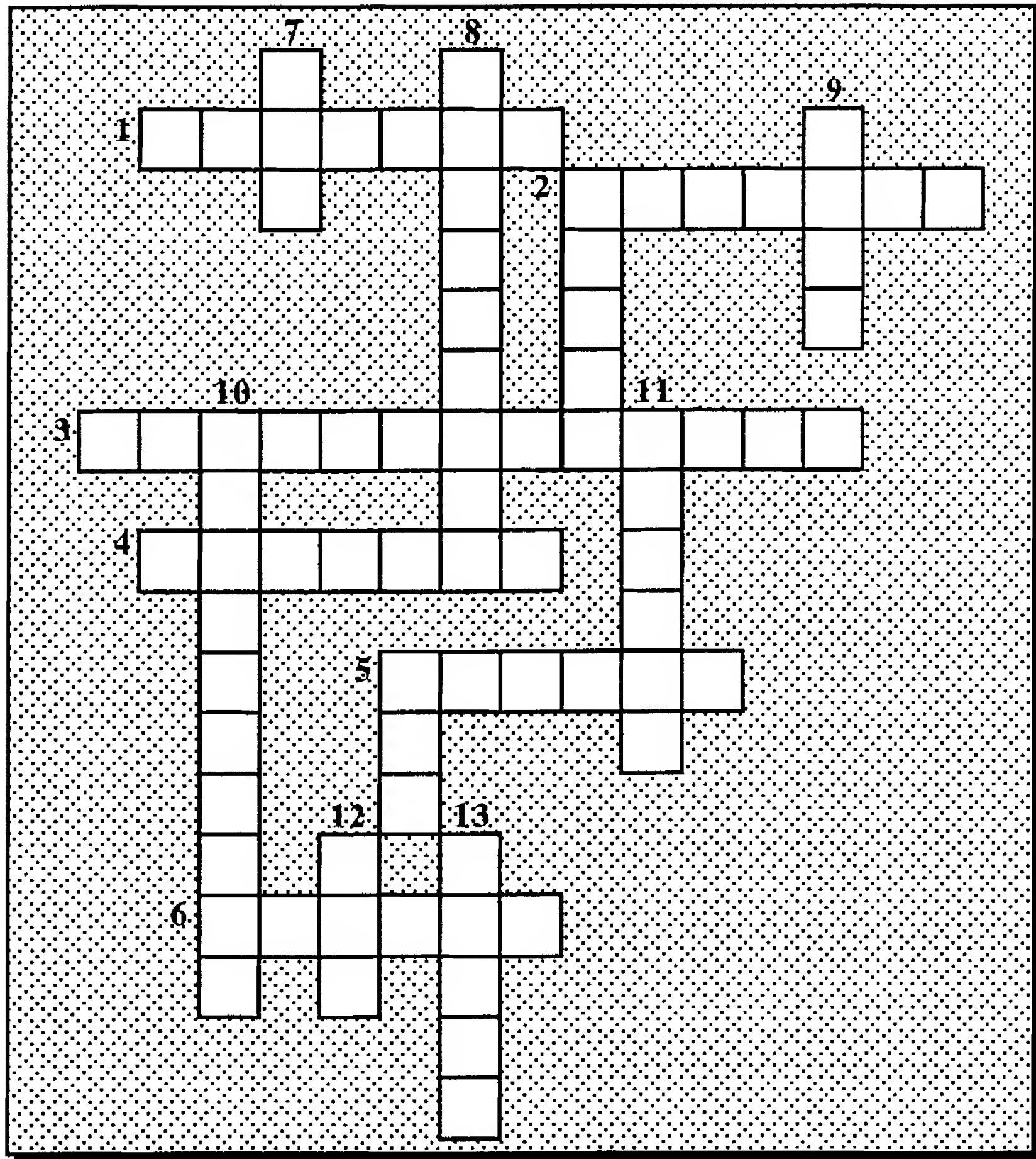
Crossword Clues

Across

1. In the expression P^.value, P must be of this type.
2. This type of variable has only two possible values: true or false.
3. The sort method that selects the largest (or smallest) value on each pass.
4. Whole numbers are declared to be of this type.
5. A one-dimensional array.
6. The form of a Pascal construct can be expressed with this type of diagram.

Down

2. A compound statement starts with this.
5. The first word in a declaration block.
7. A special pointer value.
8. A procedure that calls itself is _____.
9. The operator / is used for this type of division.
10. A set of connect nodes.
11. The author of the class text.
12. The last word in a Pascal program.
13. To ensure that an argument is not changed it should be passed into this type of parameter.

10   2.   Given the following declarations,

> **var**   str1, str2, str3: string[50];
>   m, n : integer;

and the following assignment statement,

> str1 := 'The University of Saskatchewan Huskies';

what is the value of each of the following expressions?

(i)   length(str1) + pos('chew',str1) - pos('n',str1)   _____

(ii)   copy(str1, pos('sit',str1), pos('of',str1))   _____

(iii)   concat(copy(str1,25,3), copy(str1,length('Huskies')-1,4))   _____

(iv)   length(concat(copy(str1,3,4), copy(str1,27,2), copy(str1, 3,1))) _____

(v)   pos(concat(copy(str1,35,2), copy(str1,1,length('S')))   _____

   3.   Consider the following mysterious Pascal function named Foo.

```
Function Foo (A: arrayType; X: integer; i, j: integer): integer;
     {assume that A is a 1-dimensional array containing a maximum of 100
       integer values}
begin
  if i > j then
    Foo := 0
  else      if A[i] = X then
              Foo := Foo (A, X, i + 1, j) + 1
            else
              Foo := Foo (A, X, i + 1, j);
end;
```

3   a)   Suppose arrayType is defined as:

> **type**   arrayType = array[1..100] of integer;

and D is declared as:

> **var**   D:arrayType;

and contains the values: 1, 5, 6, 3, 5, 4, 3, 2, 5, 8.

What value will be returned after each of the following function calls have been made?

(i)   Foo (D, 1, 1, 10)   _____

(ii) Foo (D, 5, 1, 10)   _____

(iii)Foo (D, 7, 1, 10)   _____

3   b)   In twenty-five words or less, what does the function Foo do?

4.   The following Pascal procedure is supposed to reverse the elements in an array. ArrayType is defined as:

type arrayType = array[1..50] of integer;

Suppose C is declared as

var C:arrayType;

and contains the sequence of values: 1, 2, 3, 4, 5, 6, 7.

After a call to the procedure reverse, i.e., Reverse(C, 1, 7), C should contain the sequence 7, 6, 5, 4, 3, 2, 1.

```
procedure Reverse (var A: arrayType; X, Y: integer);
    {assume that A is a 1-dimensional array containing a maximum of 100
      integer values}
  var temp: integer;
  begin
    for temp := X to Y do
      write(A[temp]);
    writeln;
    if X < Y then  begin
      temp := A[X];
      A[X] := A[Y];
      A[Y] := temp;
      Reverse(A, X + 1, Y - 1);
      for temp := X to Y do
          write(A[temp]);
      writeln;
    end;
  end;
```

3   a)   How many times is this procedure activated on this example (Reverse(C, 1, 7)), including the initial procedure call and all recursive calls?

7   b)   What gets written out?

1 0   5.   Write a **complete Pascal program** to read in a series of integers from a file and compute the total number of factors for each integer. For example, the factors of 12 are: 2, 3, 4, 6, thus, there are a total of 4 factors (note that 1 and 12 are not factors. The program should read the values from the input file and produce a table that contains the values read from the file and the total number of factors for each value. Suppose a file contains the integer values 8, 4, 12. The output of your program should be similar to:

value	total factors
8	2
4	1
12	4

You can assume that the input file is named 'data.txt' and that each integer is stored on a new line in this file.

2   6. a) Explain why the binary search is expected to be faster than the linear search.

8   b) You have an array of strings defined as follows:

type   list: array [1..100] of string[50];
varwords: list;

Assume that 100 words of varying length have been read into this array. Give a code segment that will sort this array of strings in order of increasing length. You do not have to write a complete program but you must show the declaration for any variable that you use. (12 marks)

1 5   7. a) A grey-scale computer image is composed of pixels (points of light in a two-dimensional array).   Imagine an array with 640 columns and 480 rows (representing 640 x 480 pixels) that might correspond to a graphic image. Each item in the array (that is, each pixel value) is a positive integer representing a light intensity value for that pixel.

Image smoothing is a process of reducing the variability between the intensity of a pixel and that of its surrounding neighbours. It involves producing a new array. Smoothing can be accomplished by averaging the values "surrounding" a particular item. Each interior value in the "smoothed" array (such as pixels[24,37]) is computed as in the following example:

new_pixels(24,37) = (pixels[23,36] + pixels[23,37] + pixels[23,38] + pixels[24,36] + pixels[24,37] + pixels[24,38] + pixels[25,36] + pixels[25,37] + pixels[25,38]) / 9

Edge pixels in the smoothed array (e.g., pixels[1,345] or pixels[267,1]) are simply copied from the original array without any averaging.

pixels   col 37   new_pixels   col 37

	8	11	9
row 24	7	9	8
	6	7	9

row 24		8	

Write a procedure Smooth that takes two arrays of real numbers corresponding to grey-scale images, and returns in the second array the "smoothed" version of the first array. A sample call would be:

    Smooth (pixels, new_pixels);

2   b) Why is it necessary to put the smoothed values into a second array? Why not simply smooth the values in the same array?

2   8. a) Briefly describe the advantage of dynamic data structures.

    b) A music publishing company maintains a database of its songs as a linked list, with each node of the linked list containing the name of the song, the writer's name, and the amount of royalties owed for this song.

5   (i) Give an appropriate record definition for this application and show how you would declare the necessary variables.

1 5   (ii) One of the songwriters in the company's database, Ring Worm, has just been exposed as a fraud. In fact, all songs he claims to have written were actually written by Penny Quarters. Write a code segment to search through the database and change every mention of Ring Worm as songwriter to Penny Quarters. Assume that the linked list is pointed at by a pointer variable L and that the last node has a pointer field of nil.

* * * * * * * * * * The End! * * * * * * * * * *